FREQUENTLY ASKED QUESTIONS:

1. **List the R name for some common probability distributions.**

Distribution	R name	Additional arguments	Argument defaults
beta	beta	shape1 (α), shape2 (β)	
binomial	binom	size (n), prob(p)	
Chi-square	chisq	df (degree of freedom)	
continuous uniform	unif	min(θ_1), max(θ_2)	min=0, max=1
exponential	exp	rate($=\frac{1}{\beta}$)	rate=1
F distribution	f	df1,df2	
gamma	gamma	shape(α),scale(β)	scale=1
hypergeometric	hyper	m=r, n=N-r, k=n (sample size)	
normal	norm	mean (μ), sd(σ)	mean=0,sd=1
Poisson	pois	lambda(λ)	
t distribution	t	Df	
Weibull	weibull	Shape,scale	scale=1

2. **Write a R program to generate 100 random observations from standard normal distribution and draw histogram for the same.**
   - R code to generate 100 random observation from standard normal distribution is
     ```
     x<-rnorm(100);
     x;
     ```
   - R code to construct histogram for the generated random observations is
     ```
     hist(x)
     ```

3. **Write a R program to generate 500 random observations from exponential distribution with θ=6, hence compute mean of the generated observations.**
   - R code to generate 500 random observations from exponential distribution is
     ```
     x<-rexp(500, rate=0.6);
     x;
     ```
   - R code to compute mean of the generated samples is
     ```
     mean(x);
     ```

4. Write the steps of R program for independent t-test, assuming that the data has to be imported from excel.

Step 1: Read the data from excel saved with the name rdata (say) in E drive.
Step 2: Assign the first variable of the data to x and the second variable to y.
Step 3: Test for normality of the data by Shapiro wilks test using the R command shapiro.test()
Step 4: Test for randomness of the data by using the R command runs.test()
Step 5: Test for equality of variances by using the R command var.test()
Note : If the test in step 3 to 5 is not significant then apply t-test .
Step 6: The command for independent sample t-test is

i. Two sided alternative

```
t.test(x,y,alternative=”two.sided”,var.equal=TRUE);
```

ii. One sided alternative

```
t.test(x,y,alternative=”greater (less)”,
var.equal=TRUE);
```

5. Write the R program for Wilcoxon test, assuming that the data has to be imported from excel.

Read the data from excel saved with the name rdata (say) in E drive and then assign the first variable of the data to x and the second variable to y.
Then the R code for Wilcoxon test is:

1 . Two sided alternative

```
wilcox.test(x,y,alternative="two.sided");
```

2. Alternative mean of x greater than mean of y

```
wilcox.test(x,y,alternative="greater");
```

3. Alternative mean of x less than mean of y

```
wilcox.test(x,y,alternative="less");
```

6. Write the R code to do correlation analysis

Read the data from excel saved with the name rdata (say) in E drive and then assign the first variable of the data to x and the second variable to y.

i. The R code for Karl Pearsons correlation is cor(x,y)

ii. The R code for Spearman’s Rank correlation is

```
cor(x,y,method=”spearman”)
```

7. Write the R code to do simple linear regression analysis.

Read the data from excel saved with the name rdata (say) in E drive and then assign the independent variable of the data to x and the dependent variable of the data to y.
The R code for the simple linear regression analysis is :

```
data1<-lm(y~x, data=rdata);
summary(data1)
```

8. **With the help of an example explain how can a interval can be computed for a population mean**

Example: The data is the score of 33 random students from college of science and mathematics 84, 93, 101, 86, 82, 86, 88, 94, 89, 94, 93, 83, 95, 86, 94, 87, 91, 96, 89, 79, 99, 98, 81, 80, 88, 100, 90, 100, 81, 98, 87, 95, and 94. The population of these scores are believe to be normally distributed with 6.8 standard deviation. Determine and interpret the 95% and 99% confidence interval of the population mean.
We observe that the sample size is greater than 30 hence we apply z-test

```
scores <- c(84, 93, 101, 86, 82, 86, 88, 94, 89, 94, 93, 83, 95,
86, 94, 87, 91, 96, 89, 79, 99, 98, 81, 80, 88, 100, 90, 100,
81, 98, 87, 95, 94)

library(BSDA)
z.test(scores, sigma.x = 6.8)
```